Claims

1.   (Currently Amended) A computer readable medium having computer-executable software code carried thereon for executing on a computing device comprising a data structure comprising computer executable instructions for a method for saving state for a semantically accessible state binding method, the method comprising:

a data structure comprising:

defining a first state frame including:

        a representation of a first state of an executing program,

        comprising values of registers and heap associated with the executing program;

        and

defining a second state frame including:

        a representation of a second state of the executing program comprising state changes made by the executing program after the first state frame is created and the second state frame includes a pointer back to the first state frame;

        a third state frame including a representation of state changes made by the executing program after a fork method creates the third state frame, and the third state frame includes a pointer back to the second frame;

        wherein the executing program saves a first state of the program in the first state frame as a semantically accessible first state object, saves a second state of the program in the second state frame as a semantically accessible second state object, and then returns to the first state of the program by using the first state object;

        wherein the third state frame is empty when created by the fork method; and

        wherein, after the fork method creates the third state frame, value of a variable of the executing program can be accessed by checking, in backwards order that the frames were created, for the value of the variable in the respective frames.

2.   (Canceled)

3.   (Currently Amended) The computer readable medium of claim 2 further

comprising ~~defining~~ a fourth <u>state</u> frame which includes changes made by the executing

program after the fork method creates the third state frame and after a set method returns the executing

program to the state of the second state frame<u>, and wherein the fourth state frame includes a pointer</u>

<u>back to the second state frame</u>.

4.   (Currently Amended) The computer readable medium of claim 3 further comprising

~~defining~~ a joined state frame including a combination of state changes in the third and fourth <u>state</u>

frames.

5.   (Currently Amended) The computer readable medium of claim 3 wherein a first thread

of the executing program makes state changes copied in the second <u>state</u> frame, and a second thread of

the executing program makes state changes copied into the third <u>state</u> frame.

6.   (Original) The computer readable medium of claim 1 wherein the second state frame

includes unchanged state read from the first state frame.

7.   (Currently amended) A computerized method comprising:

receiving via an application programming interface a request to create a state save;

in response to the request, saving a first representation of a state of an executing program

comprising copying state of the program required to return to the moment the state was saved <u>as a first</u>

<u>state frame</u>;

<u>creating a blank state frame with a backward link to the first state frame as a current state</u>

<u>frame</u>;

maintaining a second representation of subsequent state comprising changes made to the state

of the executing program after the first representation <u>in the current state frame</u>;

<u>in response to a request for value of a variable after the request to create a state save, checking</u>

<u>for the value of the variable in the first state frame;</u> and

~~resetting the executing program to the saved first representation~~ <u>changing the current state</u>

<u>frame to the first state frame</u> upon receiving a state set request at the application programming

interface.

8.   (Currently Amended) A computer system comprising:

memory and a central processing unit executing,

a program including executable instructions and an evolving present state; a ~~state component~~ first frame comprising an initial representation of a prior evolving present state of the program, ~~the initial representation comprising an instruction pointer location,~~

a state component comprising a representation of the first frame as a first-class citizen,

a second frame comprising a subsequent representation of state changes made by the program since the initial representation, ~~and~~

the program including a method for returning the program state to the prior evolving present state ~~using the instruction pointer location~~ using the state component; and

the program including a method for locating a value updated during the prior evolving present state and not present in the second frame by following a back pointer from the second frame to the first frame, reading location value from the first frame and storing the location value in the second frame.

9.   (Canceled)

10.   (Original) The computer system of claim 8, wherein the state component includes a fork method for maintaining state for a thread spawned by the program and a forked representation of state changes made by the spawned thread of the program.

11.   (Original) The system of claim 10 wherein the state component includes a join method for joining state changes made by the forked thread back into state changes of the subsequent representation.

12.   (Canceled)

13.   (Canceled)

14.   (Canceled)

15.   (Canceled)

16.   (Canceled)

17.   (Original) A computer readable medium comprising computer executable instructions for performing the method of claim 7.

18.   (Canceled)

19.   (Currently Amended) A computer readable method comprising computer executable instructions for performing a method comprising:

    receiving a request from a method, which takes as a parameter a state object, to create a saved state of an executing model;

    saving a first representation of a state of the executing model as a first state frame;

    saving a first representation of the state frame as the state object;

    creating a blank state frame with a backward link to the first state frame as a second state frame;

    maintaining, in the second state frame, a second representation of state changes made by the executing model after the first representation as the state changes occur; and

    reinstating the executing model state to the state of the first representation using the state object.

20.   (Canceled)

21.   (Canceled)

22.-29. (Canceled)

30.   (New) The computer readable medium of claim 3 wherein the data structure further comprises a fifth state frame with a pointer to the fourth state frame, wherein the fourth state frame further comprises a reference pointer with a value indicating how many frames point back to it, and wherein if the value of the reference pointer is one, then the executing program combines the fourth state frame with the fifth state frame.

31.   (New) The computer readable medium of claim 3 wherein the fourth state frame includes a pointer to the third state frame.

32.   (New) The computerized method of claim 7 further comprising writing the value of the variable in a cache in the current state frame.

33.   (New) The computerized method of claim 33 further comprising a threshold size wherein when the cache is greater than the threshold size, the cache is purged.

34.   (New) The computerized method of claim 32 further comprising a threshold size wherein when the cache is greater than the threshold size, the last used variable is overwritten.

35.   (New) The computer readable method of claim 19 further comprising:
       A fork method which saves state of the executing program as a first state frame when invoked with the state object;
       A set method which sets the state of the executing program to the first state when invoked with the state object; and
       A join method which joins the current state of the executing program with the first state when invoked with the state object.

36.   (New) The computer readable method of claim 36 wherein the first frame and the

second frame have a creation order.

37.   (New) The computer readable method of claim 36 further comprising when a variable is accessed by the executing model, checking the state frames in opposite order to their creation order until an update for the variable is found.